Learning to Behave: Adaptive Behavior for Planetary Surface Rovers

Terry Huntsberger
Jet Propulsion Laboratory
MS 82-105, 4800 Oak Grove Drive
Pasadena, CA 91109
Terry.Huntsberger@jpl.nasa.gov

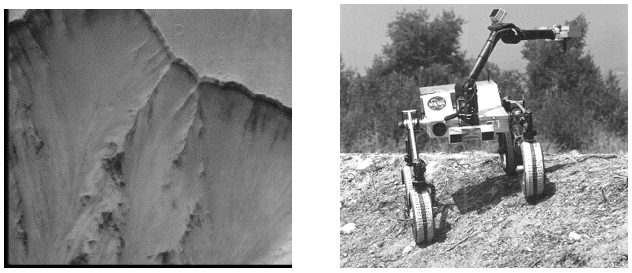
Hrand Aghazarian
Jet Propulsion Laboratory
MS 82-105, 4800 Oak Grove Drive
Pasadena, CA 91109
Hrand.Aghazarian@jpl.nasa.gov

Abstract

Robotic missions to planetary surfaces are becoming more ambitious and of longer duration. The nominal mission timeline for the MER (Mars Exploration Rovers) called Spirit and Opportunity currently on the Martian surface is 90 days, with extensions to 180 days and beyond depending on rover health. The upcoming 2009 MSL (Mars Science Laboratory) mission is planned to be 300-500 days, and will possibly involve traverses on the order of multiple kilometers. Due to time delays of up to 40 minutes round-trip for control, the rovers will require a high degree of onboard autonomous behavior that must also adapt to declining health and unknown environmental conditions during a long duration mission. This paper presents an algorithm for onboard adaptive learning of weights within a rover hierarchical behavior control framework called SMART (System for Mobility and Access to Rough Terrain). SMART is based on earlier work in free flow behavior hierarchies for planetary surface rovers (Huntsberger & Rose, 1998; Huntsberger, 2001). We also present the results of some preliminary laboratory and field studies.

1. Introduction

High-value science data acquisition on rough terrain (example shown in Figure 1(a)) is beyond the capabilities of current NASA rover designs. Although the JPL technology prototype rover SRR (Sample Return Rover) shown in Figure 1(b) has the ability to mechanically adapt itself to changing terrain by varying its shoulder angles, such an operation will require a high level of adaptability in the onboard control algorithms in order to maintain the health of the rover. In addition, as the mission progresses, the onboard control must also adapt to degraded performance due to wear-and-tear on components such as the steering and drive mechanisms.



(a) (b)

Figure 1: Planetary surface terrain and technology example for autonomous access to high risk, scientifically interesting regions. (a) Mars cliff-face with signs of water outflows; (b) JPL technology prototype of a terrain-adaptive reconfigurable rover.

We have developed a behavior-based framework called SMART (System for Mobility and Access to Rough Terrain) to address these concerns at the system level by treating rover motion, rover health, and resource management within a free flow behavior hierarchy (Huntsberger & Rose, 1998; Huntsberger, 2001). SMART uses a previously developed control architecture called BISMARC (Biologically Inspired System for Map-based Autonomous Rover Control) for long duration missions (Huntsberger and Rose, 1998; Huntsberger, 2001). It is based on a modified free-flow hierarchy (FFH) similar to the DAMN architecture (Rosenblatt and Payton, 1989; Tyrrell, 1993), and has been used successfully for a number of different simulated mission scenarios including multiple cache retrieval (Huntsberger, 1997), fault tolerance for long duration missions (Huntsberger, 1998), and site preparation (Huntsberger, *et al.*, 1999).

The major limitation in the original implementation of BISMARC in all of our previous studies was the use of fixed weights in the FFH, which effectively made it unable to adapt to situations outside of the original world model. This paper presents an onboard mechanism for learning

weights that will adapt not only to the dynamic environment around the rover, but also to the degradation of mechanical components during the mission lifetime. Underlying behaviors and the organization of the FFH are predefined based on mission requirements and rover capabilities. Our goal in this research is not to find the optimal policy, but instead one that is "good enough" to maintain rover health while still achieving its higher levels goals.

The next section briefly describes the organization of the action selection mechanism of BISMARC, followed by a discussion of the learning mechanism of the system. Next, we discuss related work, and close with experimental studies and conclusions.

2. BISMARC Organization

An example of the action selection mechanism used in BISMARC is shown in Figure 2 for a rough terrain navigation mission that is used for the experimental studies reported in this paper. The rectangular boxes represent behaviors and the ovals are sensory inputs (either fixed, direct, or derived). At the top are the high level behaviors including *Don't Tip Over, Go to Goal, Avoid Obstacles*, *Preserve Motors, Warm Up*, *Get Power,* and *Sleep at Night*. These goals are related to both task and rover safety. For example, since most planetary surface rovers have only visual sensors for navigation, the sensory input for *Proximity to Night* is derived from knowledge of the sun's position and forces the rover to sleep at night by weighting the input to *Sleep at Night* heavier (4.0) than any other behavior in the hierarchy. The *Avoid Obstacles* behavior uses the output of an onboard local navigation algorithm as recommendations for viable paths. The rovers are equipped with solar panels and the *Rest* behavior allows the batteries to recharge if the sun is up. The *Rest* behavior is also used to cool down the motors for *Preserve Motors* if they are working too hard going up a steep slope, or to stop and turn on the heaters for *Warm Up* if the internal temperature of the rover drops below a safety threshold.

The intermediate level *Change CG* behavior is an example of a sophisticated combination behavior discussed in Section 1 that works to shift the center of gravity of the rover (see Figure 1(b)) much like an animal does in response to traveling up or along a steep slope. This behavior is implemented using a finite state machine based on a well-tested algorithm for pose reconfiguration (Schenker, *et al.*, 2003a-b). The algorithm uses the onboard gyroscopes and accelerometers, which would be equivalent to the inner ear mechanism in mammals for roll and pitch determination. Recommendations for shoulder angle and arm end position changes to help stabilize the rover are generated and passed on to the bottom level behaviors.

The intermediate level behaviors are designed to interact with both the short term memory (STM), which corresponds to perceived sensory stimuli, and the long term memory (LTM), which encodes remembered sensory information. Control loops are prevented through temporal penalties (shown as T-ovals in Figure 2) that constrain the system to only repeat a behavior a predetermined number of times. The bottom level behaviors in the hierarchy fuse the sensory inputs and the activations of the higher level

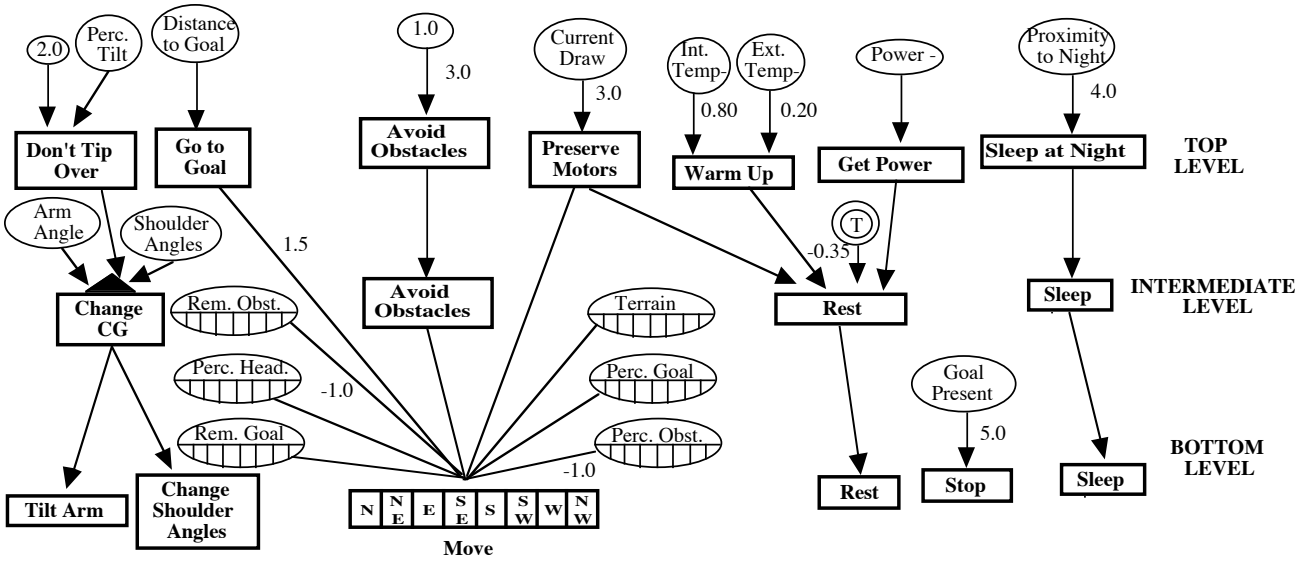


Figure 2: Free-flow hierarchy action selection mechanism for rough terrain navigation mission scenario. Ovals represent inputs derived from sensory stimuli, rectangular boxes are behaviors, and double ovals are temporal penalties. All weights on inputs to behaviors are 1.0 unless otherwise noted. Segmented boxes and ovals represent directional inputs (only cardinal directions shown but in practice continuous coverage). See text for further details.

behaviors in order to select appropriate actions for rover safety and goal achievement. The rover will continue to move until it achieves the goal position as determined by a rover localization algorithm (Hoffman, *et al.*, 1998) shown as the *Goal Present* input to *Stop* in Figure 2, or its health deteriorates due to dead batteries, freezing, burned out motors, or tipping over.

SMART's map-based LTM (Long Term Memory) is similar to hippocampus place cells. Landmarks corresponding to obstacles and goals are extensively mapped and stored for comparison to perceived inputs, with a probabilistic update of memories based on the positional variance of the rover and the match strength of the current perception to memory contents. A LTM landmark is encoded as a four-byte field that includes relative height of the landmark (2 bytes), actions leading to the landmark (1 byte), and accelerometer readings on the robot (1 byte). A similar approach is the coupled goal/representation framework of (Mataric, 1992; Mataric, 1997a). Another alternate approach is an occupancy grid that gives dense coverage of the environment, but doesn't scale well for long duration planetary surface missions (Elfes, 1987).

3. Learning Mechanism

Learning mechanisms for planetary surface rovers have the same requirements as terrestrial robots (Mahadevan & Connell, 1992): (1) noise immunity, (2) fast convergence, (3) incrementality (improving performance while learning), (4) tractability (iterations of algorithm doable in real-time), and (5) groundedness (information limited to onboard sensors). In particular, the fast convergence and tractability requirements are key for planetary surface rovers because they are typically computationally challenged (i.e., MER uses a 27Mhz CPU) due to power constraints. We address (2) and (4) through a behavior decomposition process similar to the use of *heterogeneous reward functions* developed by Mataric (Mataric, 1997b). We give the details of the reward function for updating the weights for the *Move* behavior (see Figure 2) in this section. For point (3) we use the W-learning algorithm of Humphrys (Humphrys, 1997) supplemented with a dynamic reward function directly related to rover health. For (1) we use a sequence memory similar to that of McCallum (McCallum, 1995-96) and Michaud and Mataric (Michaud & Mataric, 1998-99). Finally, we restrict our inputs to onboard sensors only as stipulated in point (5).

The weights on the links between modules are usually heuristically set based on mission goals. These goals are specified at a relatively high level without complete knowledge of the operating environment of the rover. There is however a priority derived from mission risk mitigation requirements explicitly included in the relative size of the weights. The maximum activation of the high level behaviors are weighted to give the highest priorities to rover preservation. In order of highest priority to lowest these are *Sleep at Night*, *Avoid Obstacles*, *Preserve Motors*, *Don't Tip Over*, *Get Power*, and *Warm Up*. In addition, rover health will degrade as the mission progresses, and weights chosen at full health may no longer be appropriate. Rover health is defined in Equation (1) as:

$$\text{rover\_health} = \left[w_p \,\text{power} + w_{mc} (1 - \text{motor\_current}) \right] \left[\frac{(\text{AGE\_MAX} - \text{age})}{\text{AGE\_MAX}} \right] \qquad (1)$$
$$w_p + w_{mc} = 1$$

where *power* is the current battery levels, *motor_current* is the current draw on the motors, AGE_MAX is the maximum expected lifetime for the rover, age is the current age of the rover, and w_p and w_{mc} are weights (currently both set to 0.5 since dead batteries are as lethal as burned-out drive motors). A dynamic reward function is defined in SMART based on changes in rover health and progress towards goal achievement for each step:

$$\text{reward} = w_{rh} \Delta \text{rover\_health} + w_{ga} \Delta \text{goal\_achievement} \qquad (2)$$
$$w_{rh} + w_{ga} = 1$$

where Δ is the change, and w_{rh} and w_{ga} are weights (currently set to 0.65 and 0.35 based on the relative importance of health and goal achievement determined experimentally).

Learning is only enabled in the weights on the links feeding into the *Move* behavior at the lowest level in the FFH shown in Figure 2. This is done in order to maintain the rover safety embodied in the relatively high priorities of the *Sleep at Night*, *Get Power*, and *Warm Up* high level behaviors. A modified version of the W-learning algorithm of Humphrys (Humphrys, 1997) is used in SMART to dynamically update the weights. In W-learning, agents suggest their actions with a weight W and the maximum weight is chosen as the leader. In our case there are three behaviors vying for control of *Move*, these being *Go to Goal*, *Avoid Obstacles*, and *Preserve Motors*. W-learning uses the difference between the predicted reward $\mathcal{P}$ and the actual reward $\mathcal{A}$ to determine which weights are to be updated (Humphrys, 1997).

Humphrys used a genetic algorithm run off-line to determine his reward functions. We instead use the expression in Equation (2) in order to capture the true change in the rover health through an action (the motor currents and battery levels are read in real-time). Rover behavior is extensively studied prior to launch through both laboratory and field trial studies, so the predicted changes in rover battery levels and motor currents are

known for rover movement and in fact are used for resource management planning during the missions.

A small sampling of the predicted rewards are shown in Figure 3 for typical rover behavior. The reward (1) for movement towards the goal on even terrain from a start position is the highest since rover health has a minimal change compared to progress towards the goal. As progressively steeper slopes are attempted, the rewards (2-3) start out being positive since progress towards the goal is still outweighing the impact on rover health, but become more and more negative (4-5) as the steepness increases. Backing-off the slope has a negative reward (6-7) for the relatively benign slopes since the rover health improvement in rover health is outweighed by the lack of progress towards the goal, becoming positive (8-9) for the steeper slopes. The reward (10) for driving sideways is slightly negative since minimal impact on the rover health is outweighed by the movement away from the goal. The reward is less negative than that associated with driving up the steeper slopes which will come into play during the experimental studies. The last reward (11), that of driving away from the goal is a large negative value primarily due to the movement in a direction totally opposite the goal.

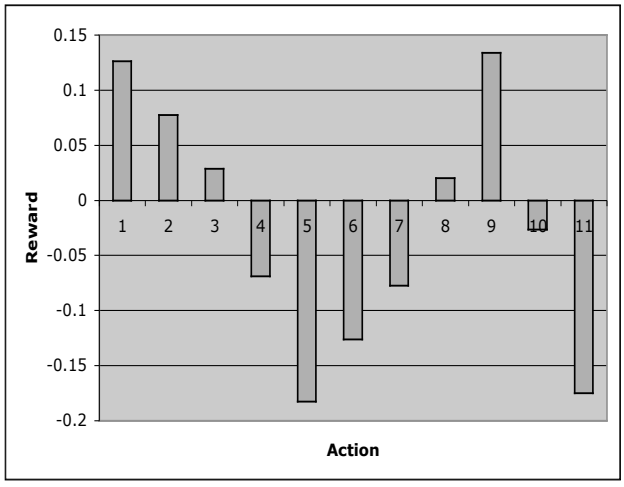


Figure 3: Sample of rewards derived using Equation (2) for a number of different types of actions (in reality the action space is continuous). Reference point is rover health of 1.0 except in cases where the rover is reacting to a current situation (i.e., backing off climbing a slope). All moves are with respect to the goal actions are: (1) normal driving on even terrain, (2) driving up a 5° slope, (3) driving up a 10° slope, (4) driving up a 25° slope, (5) driving up a 45° slope, (6) backing-off a 5° slope drive, (7) backing-off a 10° slope drive, (8) backing-off a 25° slope drive, (9) backing-off a 45° slope drive, (10) driving sideways, and (11) driving backwards. The weights were both set to 0.5 in Equation (1) and to 0.65 and 0.35 respectively for the change in rover health and change in relation to goal in Equation (2).

The *Move/Tilt Arm/Change Shoulder Angles*, *Rest*, *Stop*, and *Sleep* actions at the lowest level in the FFH shown in Figure 2 are mutually exclusive and the action with the maximum activation is chosen using a competitive action selection. The *Tilt Arm, Change Shoulder Angles,* and *Move* actions at the lowest level in the FFH shown in Figure 2 can be done simultaneously, so they are treated as a unit during the action selection process. However, progress towards the goal will be compromised if the rover tips over, so there is a dynamic relationship between the two higher level goals of *Go to Goal* and *Don't Tip Over*. The W-learning algorithm is applied to the links feeding into the *Move* behavior in the hierarchy with a time delay between activations. The *Don't Tip Over* behavior activation occurs in the first time slice, followed by the *Go to Goal* weight updates and activation. This maintains the rover health, while at the same time making progress towards the goal. Another instance where this process is applied is the relative direction that the rover moves. In order to *Preserve the Motors*, the rover will attempt to climb a steep incline, and either back off, go sideways, or rest if the perceived motor currents in the rear wheels are too high. If the weights are not dynamically adjusted, this could lead to dithering where the rover attempts to climb, backs off, and then attempts to climb in the same direction. Adaptive weighting using the W-learning algorithm changes the direction of attack, since progress towards the goal is being compromised by the dithering. For this situation, there is a time delay between application of W-learning to the two incoming links of *Go to Goal* and *Preserve Motors*, with *Preserve Motors* occurring first, followed by *Go to Goal*.

Although our convergence times are typically within 500ms, it is still desirable to limit CPU cycles devoted to learning if it may not be needed. Noise in the sensors can lead to state aliasing where the same sequence of state transitions experienced previously is not recognized. One possible solution to this problem is to provide a memory to the system (McCallum, 1995-96, Michaud & Mataric, 1998-99). We maintain a fixed number of memory traces (currently 100) of limited length (currently 25 steps) of the most recent experiences of the rover. As new experiences come in they are checked for similarity to previous sequences and merged. In the event that the behavior sequence is new, the oldest traces are deleted. These traces are organized using the tree structure developed by Michaud and Mataric (Michaud & Mataric, 1998-99). Rather than use these traces to trigger alternate behaviors as done by Michaud and Mataric, we instead use them to seed the W-learning process with the sequence of expected rewards. In our preliminary studies, we have seen a speedup of a factor of two in our step-wise learning.

4. Related Work

Prior research by Tyrrell (Tyrrell, 1993) and Bryson (Bryson, 2000) demonstrated superior performance of a

hierarchical system for action selection over purely reactive systems. In particular, the agents in the Edmund system of Bryson (Bryson, 2000) are built as related sensing and action functions that exhibit selective attention with the payoff of a higher efficiency than the modified Rosenblatt and Payton (RP) mechanisms of Tyrell (Tyrrell, 1993). A comprehensive overview of action selection systems can be found in Bryson (Bryson, 2001). Although BISMARC uses the modified RP mechanisms, the nodes in the free flow behavior hierarchy perform operations that are more sophisticated than simple combination. In some sense, they are closer to the *competence* structures of Bryson (Bryson, 2000), in that a collection of plan elements are organized as a prioritized finite state machine whose outputs converge on a specific goal. These nodes have undergone extensive evaluation at the modular level either through field or mission testing.

To date, there has been very little research into learning for hierarchical action selection systems which are typically characterized by multiple, possibly conflicting goals. The dominant learning strategy for single goal achievement such as robotic navigation has been reinforcement learning (RL), an unsupervised method that seeks to maximize a reward signal based on the utility of pairings of input and output states and their subsequent actions (Kaelbling, 1993; Kaebling, *et al.*, 1996; Sutton and Barto, 1998). One of the most popular RL algorithms is Q-learning (Watkins, 1989) and its variations such as Q-PSP (Horiuchi, *et al.*, 1996), and hierarchical Q-learning (Lin, 1993). RL algorithms typically suffer from slow convergence, large state spaces, and difficulties in handling uncertain sensory inputs. Continuous valued versions of the Q-learning algorithm have been developed to address the large state space problem (Gaskett, *et al.*, 1999; Takahashi, *et al.*, 1999; Takeda, *et al.*, 2000). These works used a continuous Q-value derived from neural networks or other function approximation methods. The state space concerns were also addressed for deterministic environments using a forgetting mechanism in a penalty-based hierarchical Q-learning algorithm, which reduces the amount of state information that an agent must maintain by using a low level agent to maintain local state information and a high level agent to maintain global state information (Yen, *et al.*, 2001; Yen and Hickey, 2002). During planetary surface rover operations, the prediction of a state following an action is difficult since it is closer to a non-deterministic process due to interactions with the terrain. Most of the RL studies to date have been confined to simulations and interior navigation in 2-D environments.

An alternate learning system that performs in the presence of a multiple conflicting goals where subtasks are only partially satisfied (Maes, 1991) is W-learning and its variations, which are based on compromise or negotiated decision making between agents (Humphrys, 1997). W-learning is a memory efficient method that is more suited for operation onboard planetary surface rovers than traditional or hierarchical Q-learning systems, and a temporally prioritized modification of it is running under SMART.

5. Experimental Studies

In order to determine the utility of SMART for planetary surface operations in rough terrain, we have run three different types of experimental studies: (1) 2000 simulated rough terrain navigation missions, (2) 50 laboratory sequences with SRR, and (3) 4 sequences with SRR in natural terrain in the Arroyo Seco outside JPL. We have attempted to match the fidelity of the simulation models for terrain and rovers to those used for the laboratory and field studies.

5.1 Simulation studies

The first series of experimental studies used simulated terrain based on MOLA (Mars Orbiter Laser Altimeter) data from the Dao Valis region of Mars, which had slopes of up to 65°. A 200 meter by 200 meter sub-area of the rough terrain dataset is shown in Figure 4 and a view of

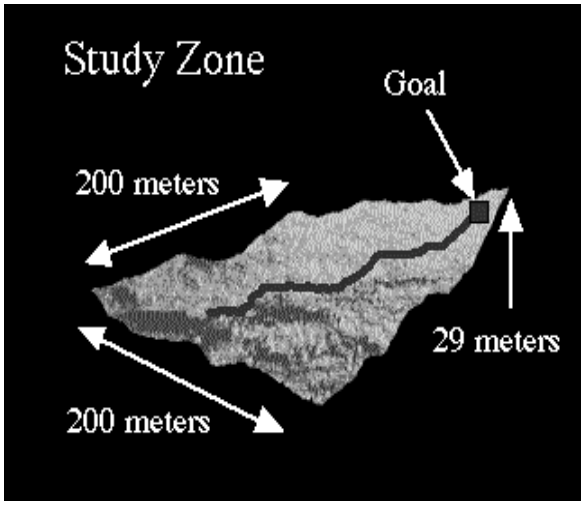


Figure 4: Terrain model for rough terrain navigation mission, with goal position at box and path shown as solid line. Study zone is 200 m by 200 m and terrain variation is from 271 m to 300 m.

the SRR during one of the simulation runs is shown in Figure 5. Mission success was defined as the attainment of the randomly selected goal position without dying due to freezing, dead batteries, burned out motors, or tipping over. The experimental setup included:

- Random starting and goal positions
- Timestep of 0.1s
- 10% loss of traction in rocky terrain
- 1 sq. km study area (5 cm resolution)
- Top speed of 15 cm/sec

The model of SRR matches the physical platform and has two sets of stereo cameras, one body-mounted and one

mast mounted, a 3 DOF (degrees of freedom) manipulator and a twelve week battery lifetime supplemented with solar panels.

Figure 5: The SRR climbing a 35° slope in simulated terrain derived from MOLA data in the Dao Valis region of Mars. The model of the rover contained full kinematics and dynamics and used a probabilistic slip assumption. The FFH shown in Figure 2 was used for control and adaptive learning for 2000 simulation runs.

Our studies had a 95.9% mission success with the onboard adaptive learning mechanism, and a 43% success rate without the adaptive learning. The primary failure mode (3.8%) for the system with learning enabled was dead batteries which from a mission standpoint would indicate a need for larger solar panels. An analysis of the 57% of the missions that failed with no learning enabled gives:

- Tipping over - 27%
- Dead batteries - 15%
- Burned out motors - 9%
- Freezing - 6%

Since 27% of the missions failed due to tipping over, the initial weights for inputs to *Move* were set too high, giving an overall bias to the *Get to Goal* behavior over rover safety related behaviors such as *Don't Tip Over*.

5.2 Laboratory studies

The second set of experimental studies was run in the Planetary Robotics Lab (PRL) at JPL and used the JPL technology prototype rover SRR shown in Figure 6. SRR has independently articulated shoulders which allow it to dynamically change its pose and lean much like an animal does on sloped terrain. The full range of shoulder movement is shown in Figure 6. SRR also has independent four wheel drive and independent four wheel steering enabling it to travel sideways.

One of the experimental runs is shown in Figure 7, where we have set up a worse case scenario of opposing hills and valleys for the rover. The SRR (Sample Return Rover) successfully negotiated the course based on a subnet of the full hierarchy shown in Figure 2. This subnet included the *Don't Tip Over, Go to Goal, Avoid Obstacles*, and *Preserve Motors* top level nodes. The *Warm Up*, *Get Power,* and *Sleep at Night* top level node activation levels were all set to zero since the interior of the lab was warm

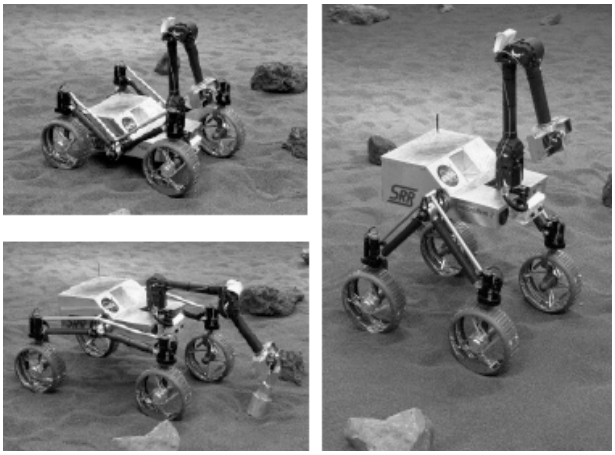


Figure 6: Sample Return Rover (SRR) range of hardware adaptation including clockwise from upper left - the lowest range of the shoulder articulation, the highest range of shoulder articulation, and the mid-range of shoulder articulation coupled with extended arm movement.

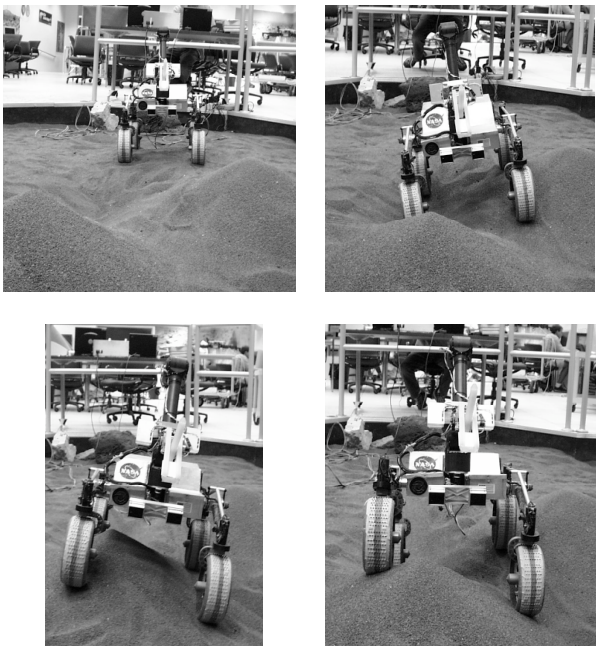


Figure 7: Clockwise from upper left: SRR performing continuous pose reconfiguration using its adjustable shoulders during a traverse in the Planetary Robotics Lab at JPL. The terrain was a set of two opposing hills and valleys, with 45º degree slopes.

and not exposed to the sun.

Another series of laboratory trials used a ramp set at a 65° slope with the rover positioned at the bottom. The goal position was on the other side of the ramp which was beyond SRR's stability capabilities to climb even with shoulder reconfiguration. Initially the rover attempted to climb the slope, but repeatedly backed off and then tried again. This behavior can be traced to the combination of *Go to Goal, Avoid Obstacles,* and *Preserve Motors* using the default weights. The learning algorithm progressively reduced the *Go to Goal* weight from 1.5 to 0.45 while at the same time increasing the weights of *Go to Goal* and *Avoid Obstacles* from 1.0 to a high-water mark of 1.6, which caused the rover to try to skirt the ramp by moving sideways while still maintaining movement towards the goal. Although adaptation of the *Avoid Obstacles* weights lagged behind those of the *Preserve Motors*, the ramp was eventually seen as an obstacle and the obstacle avoidance behavior kicked in. As the rover cleared the side of the ramp it then started movement towards the goal due to the *Go to Goal* behavior output dominating the inputs to *Move* without any obstacles or sloped terrain in front of the rover.

The dynamic weight adaptation seen in the ramp trials is shown in Figure 8, where the weights are shown for the *Go to Goal*, *Avoid Obstacles*, and *Preserve Motors* behaviors. There are rapid changes in the weights as the rover attempts to climb the ramp, followed by oscillations about a fixed point after numerous backing-off behaviors and then skirting the edge of the ramp. The variability in the weights over the trials is greatest when they are stabilizing to their new values (as seen in the size of the error bars). The eventual outcome of the sequence was that the rover learned to treat steeply sloped terrain as an obstacle, while at the same time trying to prevent motor burn-out. In the field, this behavior would be equivalent to the rover trying to find a safe way up a slope to get to the goal, as will be shown in the next sub-section.

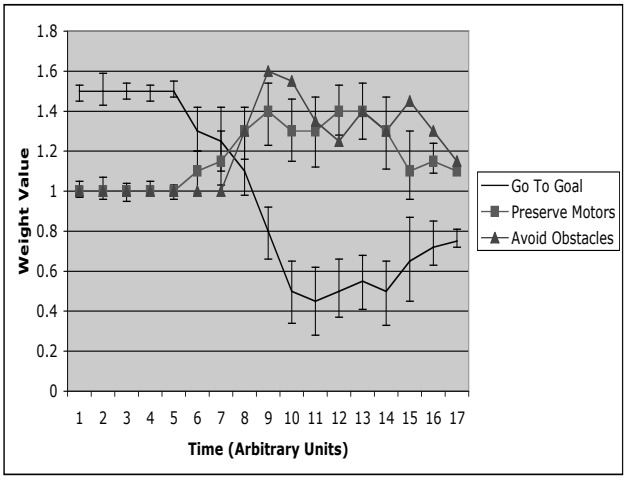


Figure 8: Adaptive learning of weights in the FFH for a rover attempting to go to a goal blocked by a steep ramp. Referring to Figure 2, the inputs to *Move* are *Go To Goal* (plain line), *Preserve Motors* (square line), and *Avoid Obstacles* (triangle line). Points on curves are the average over 20 trials with error bars giving the range of variation (error bars on *Avoid Obstacles* curve omitted for clarity).

5.3 Field studies

The last series of experimental studies was done in the Arroyo Seco, a dry wash that is next to JPL. This site is used for technology prototype rover testing and is characterized by a mixture of benign sand and rocky beds that have been scoured by the periodic water flow bordered by steeply sloped cliffs. An example of the terrain with SRR during a traverse is shown in Figure 9. The learning component of SMART was not fully implemented at the time, so only qualitative results are available at this time.

Figure 9: SRR in the bottom of a rock-strewn gully in the Arroyo Seco outside of JPL. The right shoulder is almost horizontal compared to the left one because the rover just came off of the rock behind the right rear wheel.

We were only able to complete a preliminary series of 4 runs in the Arroyo Seco and will return for more data collection in the spring of 2004 after the winter rains. An example of the skirting behavior along a slope, as previously seen in the laboratory studies discussed in section 4.2, is shown in Figure 10, where the rover approaches the slope in the left frame and is not able to climb, skirts to the side in the middle frame, and finally gets enough traction to climb to the top of the rise and continue on towards the goal.

6. Conclusions

We have developed an autonomous rover control system called SMART for planetary rovers traversing rough and highly sloped terrain. It is based on the previously developed free flow hierarchical action selection of

Figure 10: Skirting behavior of SRR along the length of a slope in the Arroyo Seco wash outside of JPL where the rover initially can not get enough traction to climb so the direction of travel favors lateral motion. Time flows from left to right in the sequence with the left frame being the initial approach almost parallel to the slope, the middle frame showing a change in rover heading more perpendicular to the slope for better traction on the top of the rise, and the right frame showing the rover continuing along its initial heading towards the goal. The mast was fixed in its orientation for this run and would have given the rover more traction if pointed up slope.

BISMARC, coupled with an onboard learning mechanism for changing weights in the hierarchy. The learning mechanism enabled SMART to maintain rover health in both simulated and actual rover studies in rough terrain. Of particular importance for future NASA rover missions was the analysis of the rover failures, indicating that an additional 52.9% of missions would potentially be successful with adaptive learning. We are currently optimizing the memory trace implementation and preparing for further trials in the Arroyo Seco (results should be collected in time for the meeting). We are also starting the integration of the SMART control techniques into the recently developed CAMPOUT (Control Architecture for Multi-robot Planetary Outposts) running on two technology prototype rovers at JPL (Huntsberger, *et al.*, 2003; Schenker, *et al.*, 2003a).

7. Acknowledgments

The authors would like to thank the three anonymous reviewers for their helpful comments on the initial draft of this manuscript. The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The views contained herein are those of the authors and are not meant to in any way imply that they are those of NASA or JPL.

References

Bryson, J.J. (2000). Hierarchy and Sequence vs. Full Parallelism in Action Selection. In *From Animals to Animats 6, Proc. of the Sixth Intern. Conf. on Simulation of Adaptive Behavior (SAB'00),* 147-156.

Bryson, J.J. (2001). *Intelligence by Design*. Ph.D. dissertation, Dept. of Elec. Engineering and Computer Science, MIT, Cambridge, MA, USA.

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3): 249-265.

Gaskett, C., Wettergreen, D., and Zelinsky, A. (1999). Q-learning in continuous state and action spaces. In *Proc. 12th Australian Joint Conf. on AI*, Sydney, Australia.

Hoffman, B.D., Baumgartner, E.T., Huntsberger, T., and Schenker, P.S. (1999). Improved rover state estimation in challenging terrain. *Autonomous Robots,* 6(2): 113-130.

Horiuchi, T., Fujino, A., Katai, O., and Sawaragi, T. (1996). Q-PSP Learning: An exploitation-oriented Q-learning algorithm and its applications. In *Proc. IEEE International Sympos. on Evolutionary Computation*, 76-81.

Humphrys, M. (1997). *Action Selection Methods using Reinforcement Learning*. PhD thesis, University of Cambridge, Cambridge, UK.

Huntsberger, T.L. (1997). Autonomous multi-rover system for complex planetary surface retrieval operations. In *Proc. Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, SPIE Vol. 3209, 220-229.

Huntsberger, T.L. (1998). Fault-tolerant action selection for planetary rover control. In *Proc. Sensor Fusion and Decentralized Control in Robotic Systems*, SPIE Vol. 3523, 150-156.

Huntsberger, T.L. (2001). Biologically inspired autonomous rover control. *Autonomous Robots,* 11(11): 341-346.

Huntsberger, T.L., Mataric, M.J., and Pirjanian, P. (1999). Action selection within the context of a robotic colony. In *Proc. Sensor Fusion and Decentralized Control in Robotic Systems II*, SPIE Vol. 3839, 84-91.

Huntsberger, T., Aghazarian, H., Cheng, Y., Baumgartner, E.T., Tunstel, E., Leger, C., Trebi-Ollennu, A., and Schenker, P.S. (2002). Rover Autonomy for Long Range Navigation and Science Data Acquisition on

Planetary Surfaces. In *Proc. 2002 IEEE International Conf. on Robotics and Automation (ICRA2002)*, Washington, DC, 3161-3168.

Huntsberger, T., Pirjanian, P., Trebi-Ollennu, A., Nayar, H.D., Aghazarian, H., Ganino, A., Garrett, M., Joshi, S.S., and Schenker, P.S. (2003). CAMPOUT: A Control Architecture for Tightly Coupled Coordination of Multi-Robot Systems for Planetary Surface Exploration. *IEEE Trans. Systems, Man & Cybernetics, Part A: Systems and Humans, Special Issue on Collective Intelligence,* 33(5): 550-559.

Huntsberger, T.L. and Rose, J. (1998). BISMARC. *Neural Networks*, 11(7/8): 1497-1510.

Kaelbling, L.P. (1993). *Learning in Embedded Systems*. MIT Press, Cambridge, MA, USA.

Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996) Reinforcement Learning: A survey. *J. of Artificial Intelligence Research*, 4: 237-285.

Lin, L.-J. (1993) *Reinforcement Learning for Robots Using Neural Networks*. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA.

Maes, P. (Ed.) (1991). *Designing Autonomous Agents Theory and Practice from Biology to Engineering and Back*, MIT Press.

Mahadevan, S. and Connell, J. (1992) Automatic Programming of Behavior Based Robots Using Reinforcement Learning. Artificial Intelligence, 55: 311-365.

Mataric, M.J. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8(3): 304-312.

Mataric, M.J. (1997a). Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Software Architectures for Physical Agents*, 9(2-3): 323-336.

Mataric, M.J. (1997b) Reinforcement Learning in the Multi-Robot Domain. *Autonomous Robots,* 4(1): 73-83.

McCallum, A.K. (1995). *Reinforcement Learning with Selective Perception and Hidden State*. PhD dissertation, Department of Computer Science, Univ. of Rochester.

McCallum, A.K. (1996). Learning to Use Selective Attention and Short-Term Memory in Sequential Tasks. In *From Animals to Animats 4, Proc. Fourth International Conf. on Simulation of Adaptive Behavior*, *(SAB'96)*, Cape Cod, MA.

Michaud, F. and Mataric, M.J. (1998) Learning from History for Behavior-Based Mobile Robots in Non-stationary Conditions. Joint Special Issue on Learning in Autonomous Robots, *Machine Learning,* 31(1-3): 141-167, and *Autonomous Robots,* 5(3-4): 335-354.

Michaud, F. and Mataric, M.J. (1999) Representation of behavioral history for learning in nonstationary conditions. *Robotics and Autonomous Systems,* 29: 187-200.

Pirjanian, P. (1998). *Multiple Objective Action Selection and Behavior Fusion Using Voting*. PhD dissertation, Laboratory of Image Analysis, Department of Medical Informatics and Image Analysis, Aalborg University, Denmark.

Pirjanian, P. and Mataric, M., (2000). Multiple objective vs. fuzzy behavior coordination. In *Lecture Notes in Computer Science on Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, D. Drainkov and A. Saffiotti, (Eds.), Springer-Verlag, 235-253.

Rosenblatt, J.K. and Payton, D.W. (1989). A fine-grained alternative to the subsumption robot control. In *Proc. IEEE/INNS Joint Conf. on Neural Networks*, 317-324.

Schenker, P.S., Huntsberger, T.L., Pirjanian, P., Baumgartner, E.T., and Tunstel, E. (2003a). Planetary Rover Developments Supporting Mars Exploration, Sample Return and Future Human-Robotic Colonization. *Autonomous Robots*, 14(2/3): 103-126.

Schenker, P.S., Huntsberger, T., Pirjanian, P., Dubowsky, S., Iagnemma, K., and Sujan, V. (2003b). Rovers for Intelligent, Agile Traverse of Challenging Terrain. In *Proc. International Conference on Advanced Robotics (ICAR'03)*, University of Coimba, Portugal, 1683-1692.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Takahashi, Y., Takada, M., and Asada, M. (1999). Continuous valued Q-learning for vision-guided behavior acquisition. In *Proc. International Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 716-721.

Takeda, M., Nakamura, T., Imai, M., Ogasawara, T., and Asada, M. (2000). Enhanced continuous valued Q-learning for real autonomous robots. In *From Animals to Animats 6, Proc. of the Sixth Intern. Conf. on Simulation of Adaptive Behavior (SAB'00)*.

Tunstel, E. (2001). Ethology as an inspiration for adaptive behavior synthesis in autonomous planetary rovers. *Autonomous Robots,* 11(11): 333-340.

Tyrrell, T. (1993). The use of hierarchies for action selection. *Journal of Adaptive Behavior*, 1(4).

Watkins, C.J. (1989). *Learning from Delayed Rewards*. Ph.D. dissertation, Cambridge Univ., Cambridge, UK.

Yen, G., Yang, F., Hickey, T., and Goldstein, M. (2001). Coordination of exploration and exploitation in a dynamic environment. In P*roc. International Joint Conference on Neural Networks (IJCNN '01),Volume 2*, 1014-1018.

Yen, G. and Hickey, T. (2002). Reinforcement learning algorithms for robotic navigation in dynamic environments. In *Proc. of the 2002 Intern. Joint Conf. on Neural Networks (IJCNN '02), Vol. 2*, 1444-1449.